OVERVIEW

The Maine Department Of Transportation uses Infrasoft's MX Professional as it's roadway design CADD application. MX uses as string-based modeling concept rather than a template-based approach used by civil design applications developed by other vendors. This document will provide detailed information on a variety of file formats that can be used to transfer design data to the Department so that it can be easily imported into our MX application.

Today's projects are designed and built by a diverse project team of Department personnel and engineering consultant firms using a variety of design applications. Data created by one team member may be used as the basis of work for another team member using a different system. The Department hopes that the emergence and acceptance of LandXML, the extensible markup language for the civil engineering CADD community, will facilitate the sharing of design data in the future, but it may take awhile for all vendors to implement LandXML import/export technology into their products. In the interim, the Department is providing information on our design application's native file formats as a medium for design data exchange.

The Data File Formats

Three data exchange formats are described in this document as well as how they can be used.

HALGN - HALGN files are used to transfer *horizontal alignment information*.

VERAT - VERAT files are MX input files used to transfer *vertical alignment information*.

GENIO - GENIO files are a GENeralized Input/Output format that can be used to *import/export MX strings other than master alignment strings*. It actually can be used to transfer alignment strings, but the Department requests that engineering design consultants use the HALGN and VERAT formats described above for that purpose. In this document, GENIO is explained with examples for the purpose of transferring *3D feature strings and cross sections*.

HALGN Format

HALGN is an ASCII format that can be used to define a horizontal alignment in MX using straight and circular elements.

A maximum of 500 elements may be processed.

Single element alignments may be defined.

Sample HALGN input data

```
MOSS
EDIT,DESIGN
004,3=MC4A
004,3=GC4A
999
HALGN,DESIGN,DESIGN
300,LB=MC4A,SC=10000.000,CF=10000.000,CE=25.000,TL=0.500
301,1,SX,X1=1074148.120202,Y1=386094.810662,X2=1074100.198409,Y2=386000.786026
301,2,LE,RA=150.000000
301,3,SX,X1=1074100.198409,Y1=386000.786026,X2=1074173.873438,Y2=385776.465497
301,4,RE,RA=150.000000
301,5,SX,X1=1074173.873438,Y1=385776.465497,X2=1074187.614075,Y2=385623.903222
301,6,LE,RA=150.000000
301,7,SX,X1=1074187.614075,Y1=385623.903222,X2=1074439.853660,Y2=384526.119739
301,8,RE,RA=675.000000
301,9,SX,X1=1074439.853660,Y1=384526.119739,X2=1074076.050658,Y2=384210.199307
999
```

Description of HALGN Format

`MOSS`

MX files begin with this line to clear any previous errors

`EDIT,DESIGN`

Tell MX to EDIT the model called DESIGN. For simplicity always use this model name in files generated from other design packages.

`004,3=MC4M`

The 004 option tells MX to delete the string labeled MC4M if it currently exists. MX alignments are named with 4 character labels beginning with "MC". The third character is selected by the user and can be any alphanumeric character. However, the selected character cannot be used for more than one alignment.

`004,3=GC4M`

Delete the corresponding Geometry String. Use the label above changing the initial character to "G"

`999`

Tell MX to end the EDIT command

`HALGN,DESIGN,DESIGN`

Begin the HALGN option. Include the model name twice.

```
300,LB=MC4M,SC=10000.000,CF=10000.000,CE=25.000,TL=0.500
```

Initiate the alignment. LB = Alignment label, SC & CF are start station. These numbers should match. CE = Station interval. Typically 10 for metric projects, 25 for Imperial. TL is a curve tolerance, use 0.5

```
301,1,SX,X1=1074148.120202,Y1=386094.810662,X2=1074100.198409,Y2=386000.786026
```

Element Data - Tangent and curve sections are defined using option 301. This line creates a Tangent from (X1, Y1) to (X2, Y2).

- The first field after the 301 record is a sequence number beginning at 1.
- The code in field 2 specifies type of element; *SX = Tangent, LE = Left-hand curve, RE = Right-hand curve.*
- X1, Y1 are coordinates at the beginning of the tangent section
- X2, Y2 are coordinates at the end of the tangent section

```
301,2,LE,RA=150.000000
```

This line creates element number 2, a left-hand curve between the tangent in the line above and the one below with a radius of 150.

```
301,3,SX,X1=1074100.198409,Y1=386000.786026,X2=1074173.873438,Y2=385776.465497
301,4,RE,RA=150.000000
301,5,SX,X1=1074173.873438,Y1=385776.465497,X2=1074187.614075,Y2=385623.903222
301,6,LE,RA=150.000000
301,7,SX,X1=1074187.614075,Y1=385623.903222,X2=1074439.853660,Y2=384526.119739
301,8,RE,RA=675.000000
301,9,SX,X1=1074439.853660,Y1=384526.119739,X2=1074076.050658,Y2=384210.199307
999
```

NOTE: When submitting horizontal alignments to the Department in HALGN format, please include a text file containing a list of special stations not included in the normal stationing interval specified in the HALGN file for each alignment. This will allow us to create an MX Master Alignment string which contains points at all stations for which cross section information might be submitted at a later date.

VERAT Format

VERAT is an ASCII format that can be used to define the vertical components of a previously created MX alignment.

Sample VERAT input data:

```
MOSS
VERAT,DESIGN,DESIGN
MC4M,10000.000000,10145.714000,7=9
10000.000000,328.220000
10003.600000,328.097000,0.030000
10004.800000,328.037000,0.030000
10010.800000,327.867000,0.030000
10041.000000,327.500000,40.000000
10095.000000,327.875477,30.000000
10118.624000,329.271000,0.030000
10135.902000,330.756000,0.030000
10145.714000,331.789000
999
```

Description of VERAT Format

MOSS

MX files begin with this line to clear any previous errors

VERAT,DESIGN,DESIGN

Begin the VERAT option. Include the model name twice.

MC4M,10000.000000,10145.714000,7=9

Begin the **profile definition** in the format:

String label, start station, end station,7=number of profile points defined

In this example, the alignment is MC4M, it begins on station 10+000, ends at station 10+145.714, and there are 7 vertical PI's between the first point and last point on the alignment. (2+7=9) There will be 9 lines required in the data block which follows this command.

10000.000000,328.220000

Data block - Start: This is the alignment start data in the format *Begin Station, Elevation*

10003.600000,328.097000,0.030000

Data block - Vertical PI Point: Vertical PI's are listed in the format *Station, Elevation, Curve Length*. In this example, the PI Station is 10+003.6, PI Elevation is 328.097, and the vertical curve length is 0.03 m. The 0.03 curve length is used to indicate a grade break. In this case the alignment is crossing another roadway at an intersection. Station 10+003.6 is the point where the alignment crosses the edge of traveled way on the intersecting road.

10004.800000,328.037000,0.030000
10010.800000,327.867000,0.030000
10041.000000,327.500000,40.000000

this line shows a vertical curve with a length of 40 at VPI station 10+041, elevation 327.5

```
10095.000000,327.870000,30.000000
10118.624000,329.271000,0.030000
10135.902000,330.756000,0.030000
10145.714000,331.789000
```

Data Block - End: This line indicates the end point on the alignment. Like the start point, the format is *End Station, Elevation*.

```
999
```

Tell MX to end the VERAT command

NOTE: Blank lines are not allowed within the VERAT data structure.

GENIO Format

GENIO is a GENeralized Input and Output option that is used to import and export model information to and from MX. GENIO format can be used to import a wide variety of string types into MX, including master alignment strings and geometry strings created from horizontal and vertical alignment definitions. The Department would prefer to receive horizontal and vertical alignment data in the HALGN and VERAT formats described earlier in this document, but feature strings and section strings can be transferred to the Department using GENIO. An explanation of the GENIO file format for 3D feature strings and section strings follows:

3D Feature Strings

This is an example of a GENIO file that will create a 3D feature string in MX.

```
MOSS
GENIO,DESIGN
017,NORM
001FORMAT(3D23.17)
003,ORDR,4=1,1,2,3,
080,CECI,7=3
0.86278740486024506D+060.23557974062420847D+060.51777335135235114D+03
0.86278725732131349D+060.23558072925923113D+060.5177803107031983 2D+03
0.86278720921827410D+060.23558172768451227D+060.5177840478596612 0D+03
0.86278726103175664D+060.23558272592411647D+060.5177856124384341 0D+03
0.86278741224405798D+060.23558371400396363D+060.5177860540562118 1D+03
0.86278766134431469D+060.23558468205148648D+060.5177864223296886 6D+03
0.86279182182447857D+060.23559013383718926D+060.5178751144059479 0D+03
0.86281114482140180D+060.23559653051477592D+060.5205170826594328 4D+03
0.86281131684491527D+060.23559658416659472D+060.5205418182099578 0D+03
0.86281591805419116D+060.23559801922184543D+060.5211332236979708 3D+03
0.86281609007772699D+060.23559807287367119D+060.5211527094160862 8D+03
0.86281706168931420D+060.23559837590624942D+060.5212622330056451 6D+03
0.00000000000000000D+000.00000000000000000D+000.5212622330056451 6D+03
999
```

A detailed explanation of each of the lines in this file follow.

MOSS

MX files begin with this line to clear any previous errors

GENIO,DESIGN

Begin the GENIO option. Include the model name that the string(s) will be created in.

017,NORM

This command changes the **Angular Input format** for the file. 017,NORM will use the system default format for MX which is typically radians. Other alternatives for this are:

DEGR - Decimal Degrees *GRAD* - Grads *QUAD* - Quads
DMS - Degrees - Minutes - Seconds (in the format D23.17) *RADI* - Radians

To specify angles in one of these other formats, substitute the appropriate Keyword for "NORM".

001FORMAT(3D23.17)

Formats The INPUT Information in the Data Block.

The format is described by a number of field descriptors separated by commas and is contained within parentheses.

A field descriptor in a format specification has the form:

[r]Cw[.d]

where **r** represents a repeat count which specifies the field descriptor is to be applied for r successive fields. The default is 1 if omitted.

C is a format code as follows:

Code	**Format**
I	Integer
A	Alpha character
X	Space
F	Real number
D and E	Double precision

w specifies the width of the field.
d specifies the number of decimal places

Example: 3D23.17 specifies that each data line will consist of 3 double-precision records representing the X, Y, and Z coordinates of each point. Each field will be 23 columns wide, and each number will have 17 decimal places.

003,ORDR,4=1,1,2,3,

Change Order - This command changes the order of the items of information in a string element. The first two dimensions of a point on a string are always Cartesian Coordinates, but the other dimensions may describe different properties of the point. In this example, the first part of the line "003,ORDR" will always remain the same. The last part of the command line indicates how the data block is organized.

4=1 indicates that 1 row of data in the data block is used to define each point. (for 3D features this is pretty straight-forward, but MX had more complex string types such as Geometry Strings that have 12-dimensional points that may be described over a number of lines.)
,1,2,3, indicates the string point dimensions the data should be assigned to. (X,Y,Z for a 3D string.)

080,CECI,7=3

String Input - This command indicates what type of string is being created. The MX string label being created in this example is **CECI**, and each point on this string will have 3 dimensions (**7=3**).

0.86278740486024506D+060.23557974062420847D+060.51777335135235114D+03
0.86278725732131349D+060.23558072925923113D+060.51778031070319832D+03
0.86278720921827410D+060.23558172768451227D+060.51778404785966120D+03
0.86278726103175664D+060.23558272592411647D+060.51778561243843410D+03
0.86278741224405798D+060.23558371400396363D+060.51778605405621181D+03
0.86278766134431469D+060.23558468205148648D+060.51778642232968866D+03
0.86279182182447857D+060.23559013383718926D+060.51787511440594790D+03
0.86281114482140180D+060.23559653051477592D+060.52051708265943284D+03
0.86281131684491527D+060.23559658416659472D+060.52054181820995780D+03
0.86281591805419116D+060.23559801922184543D+060.52113322369797083D+03

```
0.86281609007772699D+060.23559807287367119D+060.52115270941608628D+03
0.86281706168931420D+060.23559837590624942D+060.52126223300564516D+03
0.00000000000000000D+000.00000000000000000D+000.52126223300564516D+03
```

Data Block - These lines define the points for string CECI as defined in the 080 line above. Each is in the format specified in the 001Format line, 3D23.17, which is 3 fields of 23 columns in double-precision format, and 17 places to the right of the decimal point.

To end the data block defining this string's points, a final data line is added with the X and Y coordinates set to **0.000**. The data in the 3rd column of this row is of no concern other than the fact an appropriate value of the specified type must be provided. In most cases, as illustrated above, it will suffice to provide the same Z coordinate as the preceding line (the last actual point on the string.)

In the example above, a string was created that consisted of a continuous series of points. In many cases, you may want to create strings that have gaps in them (i.e. discontinuities). One such example is a string representing the top or back of curb that is broken where it crosses a driveway. To represent the point on the beginning of a discontinuity (gap), set the X value of that point to a negative value. The point representing the end of a discontinuity (gap) should have the Y value set to a negative value. (See Example 2 below).

```
999
```

999 - Tell MX to end the GENIO command.

```
GENIO,DESIGN
017,NORM
001FORMAT (3F12.3 )
003ORDR,4=1,1,2,3,
080,EH10,7=3
  862786.824  235536.820     517.519
  862786.133  235540.262     517.550
  862785.149  235545.164     517.595
  862784.165  235550.066     517.640
 -862783.690  235552.430     517.661
  862781.777 -235561.959     517.748
  862781.212  235564.773     517.773
  862780.228  235569.675     517.818
  862779.244  235574.577     517.863
       0.000       0.000     517.863
999
```

Example 2: GENIO of 3D Feature String (EH10) with a discontinuity, in 3F12.3 format

The GENIO file shown as Example 2 above is provided to illustrate the flexibility of this data format, and how a discontinuity in a string is represented. The lines:

```
 -862783.690  235552.430     517.661
  862781.777 -235561.959     517.748
```

indicate that a gap in string EH10 exists between these two points because the X coordinate is negative in the first line, and the Y coordinate is negative in the second line.

Many strings can be created from a single GENIO data file. If the model to store these strings in is the same, the format of the data block is the same, and the order and number of the string dimensions for a series of strings is the same, you need only add a new "080" command to specify the new string. An example illustrating this follows:

```
MOSS
GENIO,DESIGN
017,NORM
001FORMAT(3D23.17)
003,ORDR,4=1,1,2,3,
080,DA05,7=3
0.86274111464081192D+060.23606648830018361D+060.50539739497379867D+03
0.86274111011727946D+060.23606749373034903D+060.50533861538461605D+03
0.86274110446226096D+060.23606875065207033D+060.50526513306336307D+03
0.86274109414920362D+060.23607104290022139D+060.50513112335363013D+03
0.00000000000000000D+000.00000000000000000D+000.50513112335363013D+03
080,DB05,7=3
0.86273911359252431D+060.23606671680041630D+060.50564351031465714D+03
0.86273911013752106D+060.23606748473223695D+060.50559861538461541D+03
0.86273910448250256D+060.23606874165395825D+060.50552513306336277D+03
0.86273909510391811D+060.23607082620001762D+060.50540326606034563D+03
0.00000000000000000D+000.00000000000000000D+000.50540326606034563D+03
080,DC05,7=3
0.86273711254423880D+060.23606694530019906D+060.50598962568181770D+03
0.86273711015776265D+060.23606747573412483D+060.50595861538461571D+03
0.86273710450274416D+060.23606873265584614D+060.50588513306336353D+03
0.86273709605863050D+060.23607060950026379D+060.50577540874075561D+03
0.00000000000000000D+000.00000000000000000D+000.50577540874075561D+03
999
```

Cross Section Strings

Section information is presented as a series of offsets and elevations. For the sections to be stored as MX section strings each point requires its coordinate value. It is therefore necessary to store the master alignment from which the sections were generated and this alignment should be stored in the second named model. From this information the option determines the plan coordinates and stores the section as MX sectional strings.

MX uses a "section set identifier" to create/group sections of a particular surface together as a set. This section set identifier is a unique alphanumeric character for each surface, and is used as the first character of the section string label. The Department's standard section set identifiers are:

Template - T ***Existing Ground - G*** ***Subgrade - S***

Other section set identifiers are used on an as-needed basis to define sections for other surfaces such as ledge, underground utilities, pavement layers, etc. A standard naming convention for these additional features is not enforced.

This is an example GENIO file that creates two section strings of a proposed template (T-sections).

```
MOSS
GENIO,XSMC10,DESIGN
082,MC10,,T
  1931.490      12
           -6.700 517.120  -5.418 517.519  -5.245 517.519  -5.120 517.519
           -5.100 517.339  -3.600 517.399   0.000 517.471   3.600 517.399
            5.100 517.339   5.120 517.519   5.245 517.519   5.418 517.519
  1933.000      13
           -6.700 517.146  -5.418 517.532  -5.245 517.532  -5.120 517.532
           -5.100 517.352  -3.600 517.412   0.000 517.484   3.600 517.412
            5.100 517.352   5.120 517.532   5.245 517.532   5.418 517.532
            9.111 517.393
999999.999
999
```

In this example code, two sections are created, one for station 1+930.490, and one for station 1+933.000. An explanation of the code follows:

MOSS

MX files begin with this line to clear any previous errors

GENIO,XSMC10,DESIGN

Begin the GENIO option. Include the model name that the string(s) will be created in, then the model name where the reference master alignment string resides.

082,MC10,,T

082 - Output Standard Format Sections This command specifies the Master Alignment String associated with the sections (MC10), and the initial character for the string labels for these sections, or Section Set Identifier (T).

```
  1931.490      12
           -6.700 517.120  -5.418 517.519  -5.245 517.519  -5.120 517.519
           -5.100 517.339  -3.600 517.399   0.000 517.471   3.600 517.399
            5.100 517.339   5.120 517.519   5.245 517.519   5.418 517.519
  1933.000      13
           -6.700 517.146  -5.418 517.532  -5.245 517.532  -5.120 517.532
           -5.100 517.352  -3.600 517.412   0.000 517.484   3.600 517.412
            5.100 517.352   5.120 517.532   5.245 517.532   5.418 517.532
            9.111 517.393
999999.999
```

Section Data Block -This part of the data file contains 3 record types. **Record A** - contains the station of the section(real number with decimal point) and the number of points in the section. (Integer number, right justified). *Each section must contain from 1 to 50 points.* The data is located in columns as follows: Station (columns 1-10), Number of Points (columns 11-18). **Record B** - contains the offset and elevation values for points on the section (real number with decimal point, right justified). This data is placed in columns as follows: Off1 (col 11-18), Elev1 (col 19-26), Off2 (col 27-34), Elev2 (col 35-42), Off3 (col 43-50), Elev3 (col 51-58), Off4 (col 59-66), Elev4 (col 67-74). This format allows 4 points per line. It is not necessary to have complete lines as illustrated by Station 1933.000 above. **Record C** - This is always 999999.999, indicating the end of offset/elevation pairs for current section.

This MX option will generate a series of section strings from the sectional information which are related to the master alignment. Note that only sections whose station occurs on the master alignment string may be processed.